

[illegible]

IBM Docket No. STL920000092US1

BE IT KNOWN that We, Howard J. Glaser of San Jose, California and citizen of the United States, Laurence E. England of Morgan Hill, California and citizen of the United States, Rebecca Lau Poole of San Jose, California and citizen of the United States, and Chenhong Xia of San Jose, California and resident of the United States, have invented new and useful improvements in

of which the following is a specification:

1  
2  
3 **METHOD, SYSTEM, COMPUTER PROGRAM PRODUCT, AND ARTICLE OF**  
4 **MANUFACTURE FOR UPDATING A COMPUTER PROGRAM ACCORDING TO A**  
5 **STORED CONFIGURATION**  
6  
7  
8

9 **CROSS-REFERENCE TO RELATED APPLICATIONS**

10 Sub A7  
11 Application Serial Number \_\_\_\_\_, filed concurrently herewith on October 12,  
12 2000 for METHOD, SYSTEM, COMPUTER PROGRAM PRODUCT, AND ARTICLE OF  
13 MANUFACTURE FOR INSTALLATION AND CONFIGURATION OF A COMPUTER  
14 PROGRAM ACCORDING TO A STORED CONFIGURATION (IBM Docket  
15 STL920000062), currently co-pending, and assigned to the same assignee as the present  
16 invention; and

17 Application Serial Number \_\_\_\_\_, filed concurrently herewith on October 12,  
18 2000 for METHOD, SYSTEM, COMPUTER PROGRAM PRODUCT, AND ARTICLE OF  
19 MANUFACTURE FOR DOWNLOADING A COMPUTER PROGRAM ACCORDING TO A  
20 STORED CONFIGURATION (IBM Docket STL920000091), currently co-pending, and  
21 assigned to the same assignee as the present invention.

22 The foregoing co-pending applications are incorporated herein by reference.  
23  
24

25 A portion of the Disclosure of this patent document contains material which is subject  
26 to copyright protection. The copyright owner has no objection to the facsimile reproduction by  
27 anyone of the patent document or the patent disclosure, as it appears in the Patent and  
28 Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

## BACKGROUND OF THE INVENTION

### 1. Field of the Invention

The present invention relates in general to computer programs, and more particularly to updating a computer program.

### 2. Description of the Related Art

Customers and providers may have a number of requirements in installing or deploying an application program.

Customers may prefer to remotely install application program clients from a central server(s) in order to increase both installation productivity and maintenance productivity. Thin clients may be preferred as thin clients generally require less workstation customization, perhaps as little as a browser with the appropriate option settings and plugins. Preferably, users should be able to move between different workstations, and still be able to access their tools and data, another potential advantage of a thin client. A customer may prefer to control which application programs and which versions of the application programs are available to a particular user. In addition to controlling the application programs, a customer may wish to control access to resources and assets based upon a user's responsibilities and roles in an organization. Alternatively, the customer may wish to control and filter project work views based on the user's responsibilities and roles. In addition to such central control, the customer may want an inventory of the application programs (including versions) installed and used on a workstation or a group of workstations without the need to go to each user workstation.

1           An application program provider may also have a number of similar requirements in  
2 installing or deploying an application program. The provider may also want for licensing  
3 purposes the inventory of the application programs (including versions) installed and used on a  
4 workstation or a group of workstations without the need to go to each user workstation. To  
5 satisfy a customer's specific individual requirements, the provider would prefer a mechanism  
6 for configuring the application program to support a customer solution which may be tailored  
7 by the customer to suit the customers needs, such as role-based views.

8  
9           Conventional systems and methods have failed to provide all of the above advantages.  
10 Thus, there is a clearly felt need for a method, system, article of manufacture, and computer  
11 program product for providing remote installation and deployment with these advantages.

## SUMMARY OF THE INVENTION

The present invention comprises a method, system, article of manufacture, and a computer program product for updating an installation of an application program. A configuration of the application program corresponding to a particular user of the local application is defined and stored. In response to a user request, a determination is made that the stored configuration corresponds to the requesting user. The stored configuration is checked to determine if any items described in the stored configuration have been updated. If an item has been updated, then the updated item is retrieved, and the application program is built with the updated item. This stored configuration may also be encrypted and stored in a manifest file which may be decrypted to produce a decrypted configuration responsive to a user authentication. The decrypted configuration is checked to determine if any items described in the decrypted configuration have been updated. If an item has been updated, then the updated item is retrieved, and the application program is built with the updated item. Either or both of the manifest file and updated item may be stored on and downloaded from a remote data processing system.

One aspect of a preferred embodiment of the present invention defines a configuration of an application program corresponding to a particular user of the application program.

Another aspect of a preferred embodiment of the present invention determines that the stored configuration corresponds to the requesting user.

Another aspect of a preferred embodiment of the present invention checks the stored configuration to determine if any items described in the stored configuration have been updated.

Another aspect of a preferred embodiment of the present invention retrieves an item if the item has been updated, and the application program is built with the updated item.

1 Another aspect of a preferred embodiment of the present invention encrypts the stored  
2 configuration into a manifest file.

3  
4 Another aspect of a preferred embodiment of the present invention decrypts the  
5 manifest file in response to the user authentication.

6  
7 Another aspect of a preferred embodiment of the present invention builds the  
8 application program pursuant to the configuration decrypted from the manifest file.

9  
10 The present invention has the advantage of providing improved updating of a computer  
11 program.

12  
13 The present invention has the further advantage of providing a remote updating of an  
14 application program client from a central server(s) in order to increase maintenance  
15 productivity.

16  
17 The present invention has the further advantage of providing control over which  
18 application programs and which versions of the application programs are available to a  
19 particular user.

20  
21 The present invention has the further advantage of providing control over access to  
22 resources and assets based upon a user's responsibilities and roles in an organization.

23  
24 The present invention has the further advantage of providing control and filtering of  
25 project work views based on the user's responsibilities and roles.

26  
27 The present invention has the further advantage of providing an inventory of the  
28 application programs (including versions) installed and used on a workstation or a group of  
29 workstations without the need to go to each user workstation.

1           The present invention has the further advantage of providing for licensing purposes the  
2 inventory of the application programs (including versions) installed and used on a workstation  
3 or a group of workstations without the need to go to each user workstation.

4  
5           The present invention has the further advantage of allowing a user to configure the  
6 application program to support a customer solution which may be tailored by the customer to  
7 suit the customers needs, such as role-based views.

00607443 104300

## BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and the advantages thereof, reference is now made to the Description of the Preferred Embodiment in conjunction with the attached Drawings, in which:

**Figure 1** is a block diagram of a distributed computer system used in performing the method of the present invention, forming part of the apparatus of the present invention, and which may use the computer program product and article of manufacture comprising a computer-readable storage medium having a computer program embodied in said medium which may cause the computer system to practice the present invention;

**Figure 2** is a block diagram of an application server and a user workstation preferred in carrying out a preferred embodiment of the present invention; and

**Figure 3** and **Figure 4** are flowcharts of method steps preferred in carrying out a preferred embodiment of the present invention.



## DESCRIPTION OF THE PREFERRED EMBODIMENT

An embodiment of the invention is now described with reference to the figures where like reference numbers indicate identical or functionally similar elements. Also in the figures, the left most digit of each reference number corresponds to the figure in which the reference number is first used. While specific configurations and arrangements are discussed, it should be understood that this is done for illustrative purposes only. A person skilled in the relevant art will recognize that other configurations and arrangements can be used without departing from the spirit and scope of the invention. It will be apparent to a person skilled in the relevant art that this invention can also be employed in a variety of other devices and applications.

With reference now to the figures, and in particular with reference to **Fig. 1**, there is depicted a pictorial representation of a distributed computer system **8** which may be utilized to implement the method of, system for, article of manufacture, and computer program product of the present invention. As may be seen, distributed computer system **8** may include a plurality of networks **10** and **32**, which may be Local Area Networks (LAN), intranet networks, or internet networks, each of which preferably includes a plurality of individual computers **12** and **30**, respectively. Of course, those skilled in the art will appreciate that a plurality of Intelligent Work Stations (IWS) coupled to a host processor may be utilized for each such network.

As is common in such data processing systems, each individual computer may be coupled to a storage device **14** and/or a printer/output device **16**. One or more such storage devices **14** may be utilized, in accordance with the present invention, to store the various computer programs which may be accessed and executed by a user within the distributed computer system **8**, in accordance with the present invention. In a manner well known in the prior art, each such computer program may be stored within a storage device **14**.

Still referring to **Fig. 1**, it may be seen that distributed computer system **8** may also include multiple mainframe computers, such as mainframe computer **18**, which may be

1 preferably coupled to Local Area Network **10** by means of communication link **22**. Mainframe  
2 computer **18** may also be coupled to a storage device **20** which may serve as remote storage for  
3 Local Area Network **10** which may be coupled via communications controller **26** and  
4 communications link **34** to a gateway server **28**. Gateway server **28** is preferably an individual  
5 computer or Intelligent Work Station which serves to link Local Area Network **32** to Local  
6 Area Network **10**.

7  
8 As discussed above with respect to Local Area Network **32** and Local Area Network **10**,  
9 a plurality of server computer programs may be stored within storage device **20** and executed  
10 by mainframe computer **18**. Similarly, a plurality of client computer programs may be stored  
11 within storage devices **14** and executed by individual computers **12** such that distributed  
12 client/server computer programs are provided. Of course, those skilled in the art will  
13 appreciate that the mainframe computer **18** may be located a great geographical distance from  
14 Local Area Network **10**, and similarly, Local Area Network **10** may be located a substantial  
15 distance from Local Area Network **32**. That is, Local Area Network **32** may be located in  
16 California while Local Area Network **10** may be located within Texas and mainframe computer  
17 **18** may be located in New York.

18  
19 As will be appreciated upon reference to the foregoing, it is desirable for a user of a  
20 workstation **12** to be able to remotely install an application program from server **18**. A  
21 configuration of the application program corresponding to this particular user of the application  
22 program is defined and stored on a remote storage **20** of the remote server **18**. In response to a  
23 user request communicated from the user workstation **12** to the server **18** via LAN **10** and  
24 communications link **22**, the server **18** determines that the configuration stored on storage **20**  
25 corresponds to the requesting user. Server **18** authenticates the user, and after such  
26 authentication, data is downloaded to the workstation **12** from the remote server **18** according  
27 to the stored configuration. This stored configuration is also encrypted and stored in a manifest  
28 file which is also downloaded. Responsive to the user authentication, the manifest file is  
29 decrypted, and the application program is constructed, configured, and installed on workstation

1     **12** pursuant to the configuration decrypted from the manifest file. The user may also move  
2     from a first workstation **12** on which the application program is installed to a different second  
3     workstation **12** on LAN **10** and still be able to access the user's tools and data which may be  
4     downloaded and installed on the second workstation **12** in a similar fashion to the download  
5     and installation on the first workstation **12**. Using the user configurations and authentications  
6     stored on storage **20** and the server **18**, a customer may control the application programs, the  
7     versions of these application programs, and resources that are available to the particular user.  
8

9             The customer may also inventory the application programs (including versions)  
10     installed and used on a workstation **12** or a group of workstations **12** on LAN **10** without the  
11     need to go to each user workstation. An application program provider may also want for  
12     licensing purposes such an inventory which may be produced by customer's server **18** or the  
13     provider's server **26**.  
14

15             Referring now to **Figure 2**, a block diagram of a preferred embodiment of the present  
16     invention is illustrated. A file known as a manifest file or manifest **244** provides a description  
17     as to which plugins, configurations, and other resources should be loaded when an application  
18     program, such as an Integrated Development Environment (IDE), is initialized. Although the  
19     preferred embodiment is presented in the context of an IDE, those skilled in the art will  
20     recognize that the invention may be practiced with other types of application programs. The  
21     preferred embodiment uses this manifest to provide a mechanism for plugging in tools or  
22     components to provide a customized application program with the appropriate viewers, editors,  
23     browsers, and other resources to support the user's needs. For example, a customized IDE  
24     may be provided with the appropriate viewers, editors, and browsers to support a specific  
25     developer's needs. Furthermore, the preferred embodiment allows remote customization and  
26     configuration of the application program for an individual user based on the user's role and  
27     responsibility in an organization.  
28

29             The preferred embodiment of the present invention comprises an application server

1     **202**, which may be behind a web server, that is used to define users **204**, **206**, and **208** in a user  
2     table **210** and to define the users' application program configurations (**214**, **216**, and **218**) in a  
3     user IDE configuration table **220**. This configuration may include plugins **222** to be  
4     downloaded , user authorizations for each user **224**, templates for builds **226** (pre packaged  
5     site JCL), location/configuration of site servers **228**, other information **230** such as the location  
6     and configuration of system input/output (I/O), and any site specific documentation **232** which  
7     may be shown in a browser or plugin.

8  
9             A security system **234**, such as a certificate or simple a logon dialog, may be used by a  
10     user to initiate a session **236** with the server **202**. Upon authentication of the user by the  
11     security system **234**, a table lookup of the user table **210** is performed and plugins **222**,  
12     authorizations **224**, templates **226**, server information **228**, and documentation **232** are  
13     aggregated into the user's components **238** which are downloaded via the download **240** to the  
14     user's workstation **242**. In addition, a manifest **244** is created with this configuration  
15     information. This manifest **244** is encrypted and requires authentication from a security system  
16     **246** on the workstation **242** to be decrypted and used to build **248** an application program **250**.

17  
18             After the download **240** to the user's workstation **242**, the user through a local logon in  
19     the security system **246** invokes the decryption and use of the manifest **244** to build **248** the  
20     user's personalized application program **250**. Meanwhile a request may be sent, preferably  
21     unknown to the user, to the server **202** to check a component configuration table **252**  
22     maintained in the application server **202** to determine if any items have been updated.  
23     Alternatively, a request may be sent to the server **202** to record in a components inventory **254**  
24     which tools are being used by the user.

25  
26             Referring now to **Figures 3** and **4**, the flowcharts **300** and **400** illustrate the operations  
27     preferred in carrying out the preferred embodiment of the present invention. In the flowcharts,  
28     the graphical conventions of a diamond for a test or decision and a rectangle for a process or  
29     function are used. These conventions are well understood by those skilled in the art, and the

1 flowcharts are sufficient to enable one of ordinary skill to write code in any suitable computer  
2 programming language.

3  
4 After the start **305** of the process **300**, process block **310** defines a configuration of the  
5 application corresponding to a particular user **206** of the application, such as user 2, and  
6 process block **315** then stores the configuration **216** on a remote server **202** executing on a  
7 remote data processing system **18**. Alternatively, the configuration **216** may be stored on  
8 local data processing system **242**. Thereafter, process block **320** may initiate a session between  
9 the local data processing system **242** and the application server **202** on remote data processing  
10 system **18** in response to the user **206** requesting the application. Responsive to this user  
11 request, process block **325** authenticates the user **206**, and process block **330** then determines  
12 that the stored configuration **216** corresponds to the requesting user **206**. Process block **335**  
13 then downloads data **240** from the remote data processing system **202** to the local data  
14 processing system **242** according to the stored configuration **216**. Alternatively, process block  
15 **335** may access the data **240** from the local data processing system **242** or from another  
16 resource within the distributed computer system **8** according to the stored configuration **216**.  
17 Thereafter, processing continues to decision block **450** on **Figure 4**. This processing path is  
18 illustrated by flowchart connectors **A**, **340** on **Figure 3** and **445** on **Figure 4**. Process block  
19 **450** encrypts the stored configuration **216** into a manifest file **244** which is then downloaded by  
20 process block **455** from the remote data processing system **202** to the local data processing  
21 system **242**. Alternatively, process block **450** may encrypt the stored configuration **216** into a  
22 manifest file **244** which is then stored on the local data processing system **242**. After the  
23 download **240** of the manifest file **244** and the components **238** to the user workstation **242**, the  
24 user **206** may request a build **248** of the downloaded components **238** pursuant to the  
25 configuration in the manifest file **244**. Responsive to such a user request, the user is  
26 authenticated by process block **460**, and if the user **206** is authenticated, then process block **465**  
27 decrypts the manifest file **244**. Thereafter, process block **470** builds, installs, and configures  
28 the application program **250** pursuant to the configuration decrypted from the manifest file **244**.  
29 The process then ends at process block **475**.

1 In an alternative embodiment of the present invention, an application program is  
2 updated for execution on the data processing system. In this alternative embodiment, process  
3 block **330** also determines if any items described in the configuration have been updated. If an  
4 item has been updated, then process block **335** retrieves the updated item, and process block  
5 **470** builds the application program with the updated item.

6  
7 In still another alternative embodiment of the present invention, an application program  
8 is updated for execution on the data processing system based upon determining if any items  
9 described in the decrypted configuration have been updated. In this alternative embodiment,  
10 process block **465** also determines if any items described in the decrypted configuration have  
11 been updated. If an item has been updated, then process block **470** retrieves the updated item  
12 and builds the application program with the updated item.

13  
14 Using the foregoing specification, the invention may be implemented using standard  
15 programming and/or engineering techniques using computer programming software, firmware,  
16 hardware or any combination or sub-combination thereof. Any such resulting program(s),  
17 having computer readable program code means, may be embodied within one or more  
18 computer usable media such as fixed (hard) drives, disk, diskettes, optical disks, magnetic tape,  
19 semiconductor memories such as Read-Only Memory (ROM), Programmable Read-Only  
20 Memory (PROM), etc., or any memory or transmitting device, thereby making a computer  
21 program product, i.e., an article of manufacture, according to the invention. The article of  
22 manufacture containing the computer programming code may be made and/or used by  
23 executing the code directly or indirectly from one medium, by copying the code from one  
24 medium to another medium, or by transmitting the code over a network. An apparatus for  
25 making, using, or selling the invention may be one or more processing systems including, but  
26 not limited to, central processing unit (CPU), memory, storage devices, communication links,  
27 communication devices, servers, input/output (I/O) devices, or any sub-components or  
28 individual parts of one or more processing systems, including software, firmware, hardware or  
29 any combination or sub-combination thereof, which embody the invention as set forth in the

claims. User input may be received from the keyboard, mouse, pen, voice, touch screen, or any other means by which a human can input data to a computer, including through other programs such as application programs, databases, data sets, or files.

One skilled in the art of computer science will easily be able to combine the software created as described with appropriate general purpose or special purpose computer hardware to create a computer system and/or computer sub-components embodying the invention and to create a computer system and/or computer sub-components for carrying out the method of the invention. Although the present invention has been particularly shown and described with reference to a preferred embodiment, it should be apparent that modifications and adaptations to that embodiment may occur to one skilled in the art without departing from the spirit or scope of the present invention as set forth in the following claims.